2011 International Conference on Information Communication and Management
IPCSIT vol.16 (2011) © (2011) IACSIT Press, Singapore

Syntactic Bank-based Linguistic Steganography Approach

Ei Nyein Chan Wai, May Aye Khine

University of Computer Studies, Yangon
einyeinchanwai@gmail.com

Abstract. As information exchange plays a vital role in today people's daily activities, information security becomes more important and steganography is one of the solutions of it. In this paper, we propose a syntactic bank-based linguistic steganography approach for information security. While the input raw sentence is parsed with the Stanford parser that can produce a phrase structure of the sentence used to produce the syntax of the sentence, Shannon-Fano coding is used to compress the input secret message as minimum total bits length as possible. Then, syntax transformation task searches the syntax set of the given sentence within the syntax bank, and transforms it into a desired syntax that can represent the key-controlled semi-randomly generated secret bits intended to hide in the sentence. The resulting stego text will still be innocent-looking by applying semantically unchanged syntax transformation on the input text. Again, we intend to apply SHA 512 hash algorithm to produce keyed-hash message authentication code (HMAC) to improve robustness of resulting stego text.

Keywords: linguistic steganography, text information hiding, syntax, data compression, random number generation.

1. Introduction

Steganography means "concealed writing" to establish communication between two parties whose existence is unknown to a possible attacker. Moreover, it is the term applied to any number of processes that will hide a message within an object, where the hidden message will not be apparent to an observer. It has found usages variously in military, diplomatic, personal and intellectual property applications since historical times until the present day.

There are three dimensions in a stego system,

1. Payload Capacity : the ratio of hidden information to cover information.
2. Robustness : the ability of the system to resist against changes in the cover object.
3. Imperceptibility : the potential of the generated stego object to remain indistinguishable from other objects in the same category [3].

These are often contradictory requirements; for example, imperceptibility limits the payload.

In digital steganography for today era, modern steganography includes the concealment of information within computer files. The different types of secret message, such as audio, image, and text, can be hidden in the different types of cover media, such as audio, video, image, text, and so on. Among these different cover media, texts are widely used in several processes. However, it is also the most difficult kind of steganography because it is due largely to the relative lack of redundant information in a text file.

Text steganography is broadly classified into the two categories; linguistic approach which is the art of using written natural language to conceal secret messages and format-based approach which used physical formatting of text as a place in which to hide information. The former can be divided into semantic and syntactic method and the latter can also be divided into line-shift, word-shift, open-space and feature encoding [7].

In this paper, a steganographic approach is proposed for linguistic steganography by using the Shannon-Fano compressing algorithm, the statistical Stanford parser and a syntactic method based on the syntax bank. In addition, we apply SHA 512 hash algorithm to generate keyed-hash message authentication code (HMAC) in order to represent the identity of the resulting stego text. In section 2, a brief overview of existing linguistic steganography methods will be presented. Section 3 will explain the syntax of the language. Section 4 presents our proposed method. Finally, the conclusion and future work will be placed in section 5.

2. Linguistic Steganography

Linguistic Steganography is concerned with making changes to a cover text in order to embed information, in such a way that the changes do not result in ungrammatical or unnatural text. Most of the linguistic steganography methods use either lexical (semantic) or syntactic transformations or combination of both. The synonym substitution is the popular lexical steganography method. It substitutes the original word with another word that possesses mostly the same meaning as the original word. The syntactic methods transform the grammatical style of the original sentences. It also constitutes the swapping of word that cannot affect the meaning of the original sentence.

2.1. Lexical Steganography

In [1], the writers used synonym replacement by using a word dictionary to get synonym. Furthermore, the secret text to be hidden is first compressed by Huffman Compression Algorithm to be consumed in selection of synonyms. In [3], Brecht Wyseur, Karel Wouters, and Bart Prenee proposed a linguistic steganography based on word substitution over an IRC channel. The generation of the word substitution table is based on a session key and used synonyms from a public thesaurus.

2.2. Syntactic Steganography

According to our recent study, B. Murphy and C. Vogel mainly proposed syntactic methods for steganography. In [2], they examined two highly predictable and reasonably common grammatical phenomena in English that can be used in data hiding, the swapping of complementisers and relativisers, which rely on a well-established technology: syntactic parsing.

The other people explored the morphosyntactic tools for text watermarking and developed a syntax-based natural language watermarking scheme in [5]. The unmarked text is first transformed into a syntactic tree diagram in which the syntactic hierarchies and the functional dependencies are coded. The watermarking software then operates on the sentences in syntax tree format and executes binary changes under control of Wordnet to avoid semantic drops.

In [6], the authors developed a morphosyntax-based natural language watermarking scheme in which a text is first transformed into a syntactic tree diagram where the hierarchies and the functional dependencies are made explicit. The watermarking software then operates on the sentences in syntax tree format and executes binary changes under control of Wordnet and Dictionary to avoid semantic drops.

2.3. Combining Lexical and Syntactic Steganography

Some works in the steganography combine lexical and syntactic methods. These methods work at the sentence level to hide the intended secret information. In [9], the proposed scheme works at the sentence level while also using a word-level watermarking technique. It uses XTAG parser for parsing, dependency tree generation and linguistic feature extraction and RealPro for natural language generation.

3. Syntax of Language

The syntax of a language is the set of rules that language uses to combine words to create sentences. The parts of speech of words combine into phrases; noun phrase, verb phrase, propositional phrase, adjectival phrase, and adverbial phrase. One way of diagramming the structure of a sentence is called phrase structure rules. For example: S -> NP VP “A sentence is made up of a noun phrase and a verb phrase.”

Most of today parsers produce the above phrase structure. In subject-verb-object representation, the noun phrases in the above structure become either subject or object of the sentence. Some works have done on extraction of subject(s), verb and object(s) from a sentence’s phrase structure. In [4], extraction of subject-

predicate-object (subject-verb-object) triplets from English sentences is done by using well known syntactical parsers for English; namely Stanford Parser, OpenNLP, Link Parser and Minipar.

Moreover, a sentence is actually a clause, a set of words that includes at least a verb and probably a subject noun. But a sentence can have more than one clause. There may be a main clause (or independent clause) and one or more subordinate clauses [11]. For instance, "After we have received the goods, we will settle the account." Finally, a sentence can also have two or more main (independent) clauses, joined by coordinating conjunctions [11]. For example, "Either I go or he goes."

4. Proposed Approach

Firstly, the cover text is parsed by the parser while the secret message is compressed by the Shannon-Fano algorithm. Then, the parsed cover text sentence is transformed into one of the syntax forms within the syntax set of the original sentence. This transformed syntax is the one that has been marked with the longest binary sequence in the compressed binary form of the secret message. As long as the compressed secret message remains to hide in the cover text, the above processes are done for each of the cover text sentences. When there is no more binary sequence to hide, the cover text becomes the stego text that contains the secret in it, and ready to send over the communication channel together with the codes that compressed the secret. Moreover, the HMAC of the stego text is generated by using SHA-512 hash algorithm and the secret key that have been shared between the sender and the receiver to identify the integrity of this stego text.

When the stego text reaches to the receiver side, it is firstly checked whether the HMAC produced by the shared key is the same as the original HMAC went together with the stego text. If so, the stego text is parsed by the parser to get the grammar structure of it. Then the syntactic checking step finds the syntax set of the stego text sentence by sentence. Moreover, this step finds out the corresponding binary sequence of it. By carrying out these steps for each sentence of the stego text, the binary representation of the compressed secret message will be retained. This is then decompressed by the codes came together with it. If the HMACs are different, the receiver can suspect the integrity of the stego text. So, the current stego text is drooped and asked the sender to resend again the message.

4.1. Shannon-Fano Algorithm

This is a technique for constructing a prefix code based on a set of symbols and their probabilities. The symbols are arranged in order from most probable to least probable, and then divided into two sets whose total probabilities are as close as possible to being equal. All symbols then have the first digits of their codes assigned. As long as any sets with more than one member remain, the same process is repeated on those sets, to determine successive digits of their codes [12].

4.2. The Stanford Parser

This is a Java implementation of probabilistic natural language parsers, a program that works out the grammatical structure of sentences. It uses knowledge of language gained from hand-parsed sentences to try to produce the most likely analysis of new sentences. Although these statistical parsers still make some mistakes, but commonly work rather well [13]. The output of this parser, the phrase structure grammar representation of the sentence, is used as the input of the syntactic transformation stage of the sender and the syntactic checking stage of the receiver.

4.3. Syntactic Steganography using Syntax Bank

The proposed method uses syntax bank that consists of a number of the syntax sets and has already shared between the sender and the receiver. A syntax set is a set of all available syntax forms of a sentence which are semi-randomly assigned a binary number for each. The number of secret bits which can be hidden in a sentence depends on the number of available syntaxes in the syntax set in which the sentence's original syntax exists. If there is more than one clause in the input sentence, the syntax set includes not only the syntax for the whole sentence, but also that for each clause.

4.3.1. Key-Controlled Semi-Random Number Assignment

The sender and the receiver have already shared a key that is used as a seed to produce the unique random sequence that is assigned to the syntactic rules of the set. This algorithm can generate the random sequence without repeating. Only the sender and receiver who shared the seed can generate the random sequence of correct order. Even the intruder obtains the syntax set; it cannot be possible to assign the correct binary numbers sequence because of lack of knowledge about the seed. The algorithm that can produce the unique random numbers is described as follows.

```
function generateUniqueRandom (Long seed, int max) return random
        temp = generate new-random within 0 to max interval;
        if ( ! previous-random) add temp to previous-random;
        else {    while ( temp € previous-random)
                      temp = generate new-random;
                  }
        return temp;
```

Fig. 3: The algorithm for generating unique random number

4.3.2. Syntax Transformation

This step transforms the input sentence into the desired syntax form. The most possible transformation is active-passive transformation. This can be used for all sentences and clauses that contain subject, verb, and object. In addition, there is also possible to interchange the clauses back and front. Apart from this, there may be many other ways to transform the sentence retaining its meaning such as topicalization, adverb displacement, and so on.

4.4. SHA-512 based Keyed-Hash Message Authentication Code (HMAC)

HMAC is a mechanism for message authentication using cryptographic hash functions. HMAC can be used with any iterative approved cryptographic hash function, in combination with a shared secret key. The cryptographic strength of HMAC depends on the properties of the underlying hash function [8].

In this system, we intend to use SHA-512 hash algorithm to produce HMAC. The maximum message size of this algorithm is 2^{128}-1 bits and its block size is 1024 bits. The final result is a 512-bit message digest. It is has collision resistance strength of 256 bits, and the estimated preimage resistance strength of 512 bits [10].

4.5. Experimental Result

At the time of writing, we tested our system with 6 text files with 200 sentences as cover text. The average payload capacity is about 0.6 bits per sentence. As the hidden capacity of syntax based methods is normally between 0.5 and 1.0 bits per sentence, the capacity of our method is within the acceptable range. This payload capacity of our proposed system can be improved by adding other transformation methods. The more syntax forms we can apply to, the better the capacity of our system will be.

The imperceptibility of the proposed system is measured by the judgments of 20 people. 95% of the judgments said that the input cover text and the output stego text have the same meaning.

The robustness of the system can be achieved by applying SHA-512 based HMAC to the output stego text. Because of this HMAC, the integrity of the incoming stego text can be determined and the robustness can be improved.

5. Conclusion and Future Work

Our proposed system will not change the appearance of the cover text because it is based upon the syntax instead of the format-based method. In addition, the meaning of the result stego text sentences is the same as their original cover text sentences because the syntax set of the proposed system is a collection of different syntax forms that can produce the same meaning. Due to this retaining appearance and meaning, the proposed method can produce natural looking text as the cover text.

Furthermore, the proposed method uses the key-controlled semi-random assignment for syntax forms in the syntax set. Even though the intruder could have the syntax set, they cannot achieve the exact binary value without having the key. This improves the strength of our proposed system.

6. References

[1] A.M.nanhe, M.P.Kunjir, S.V.Sakdeo. Improved Synonym Approach to Linguistic Steganography. *http://dsl.serc.iisc.ernet.in/~mayuresh/ImprovedSynonymApproachToLinguisticSteganography.pdf*.

[2] B.Murphy, C.Vogel. The syntax of concealment: Reliable methods for plain text information hiding. In: *Proc. of the SPIE Conference on Security and Steganography and Watermarking of Multimedia Contents IX*. San José. 2007.

[3] B.Wyseur, K.Wouters, B.Preneel. Lexical Natural Language Steganography System with Human Interaction. In: *Proc. of the 6th European Conference on Information Warfare and Security*. 2007. 303-312.

[4] D.Rusu, L.Dali, B.Fortuna, M.Grobelnik, D.Mladenić. Triplet Extraction from Sentences. *10th International Multi-conference on Information Society(IS-2007)*. Ljubljana: Slovenia. 2007.

[5] H.M.Meral, E.Sevinç, E.Ünkar, B.Sankur, A.S.Özsoy, T.Güngör. Syntactic tools for text watermarking. In: *Proc. of the SPIE Conference on Security and Steganography and Watermarking of Multimedia Contents IX*. San José. 2007.

[6] H.M.Meral, B.Sankur, A.S.Özsoy, T.Güngör, E.Sevinc. Natural language watermarking via morphosyntactic alterations. *Computer Speech and Language 23*. 2009.107–125.

[7] H.Singh, P.K.Singh, K.Saroha. A Survey on Text Based Steganography. In: *Proc. of the 3rd National Conference; INDIACom-2009 Computing For Nation Development*. 2009.

[8] Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, USA. The Keyed-Hash Message Authentication Code (HMAC). *FIPS PUB 198*. 2002.

[9] M.Topkara, U.Topkara, M.J.Atallah. Words Are Not Enough: Sentence Level Natural Language Watermarking. *MCPS'06*. California: USA. 2006.

[10] Q.Dang. Recommendation for Applications Using Approved Hash Algorithms. *NIST Special Publication 800-107*. 2009.

[11] http://webspace.ship.edu/cgboer/syntax.html (see at 14.7.2011)

[12] http://www.wikipedia.org (see at 10.7.2011)

[13] http://www-nlp.stanford.edu/software/lex-parser.shtml (see at 14.7.2011)

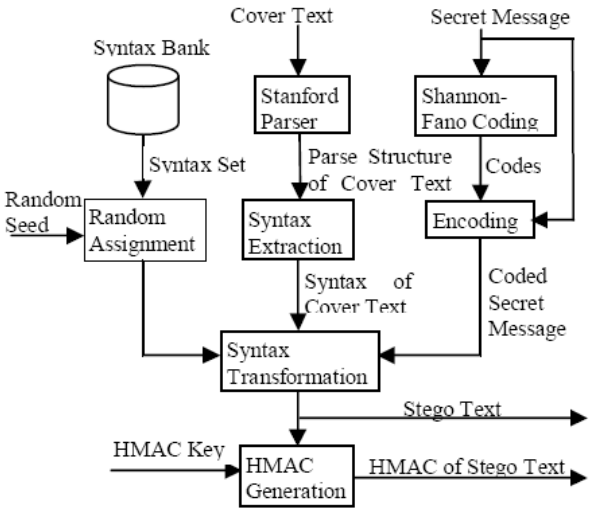

Fig. 1: Proposed System (Sender Side)

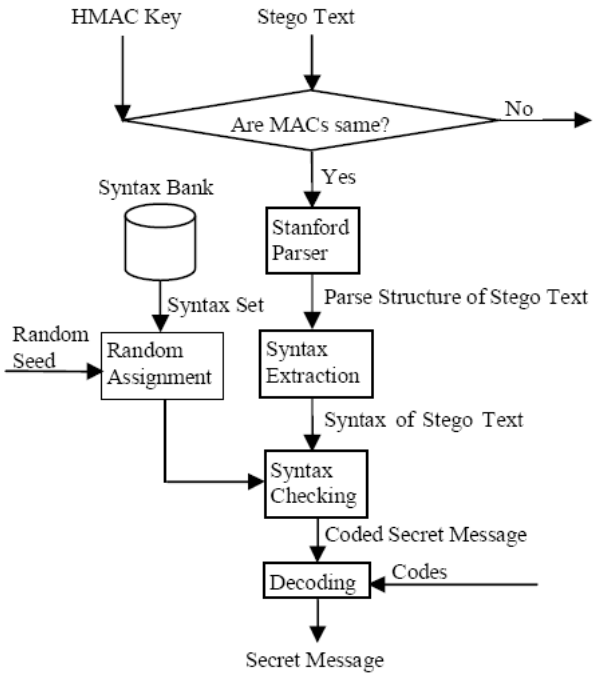

Fig. 2: Proposed System (Receiver side)